Cartes de communautés pour l'adaptation interactive de profils dans un système de filtrage d'information

An-Te Nguyen, Nathalie Denos, Catherine Berrut

Cartes de communautés pour l'adaptation interactive de profils dans un système de filtrage d'information

An-Te Nguyen* — Nathalie Denos* — Catherine Berrut*

** Laboratoire CLIPS-IMAG*
385 rue de la Bibliothèque, BP 53, 38041 Grenoble cedex
{An-Te.Nguyen, Nathalie.Denos, Catherine.Berrut}@imag.fr

Chercheurs

RÉSUMÉ. *Dans le contexte actuel de surcharge d'informations, les utilisateurs peuvent s'en remettre à des systèmes de filtrage qui leur recommandent en permanence, en se basant sur leur profil, des informations vraisemblablement pertinentes. Néanmoins, un changement dans leur besoin d'information n'est pas toujours bien pris en compte à cause du rôle relativement passif des utilisateurs dans la plupart des systèmes existants. Nous présentons dans cet article la possibilité d'utilisation interactive de « cartes de communautés » pour cette tâche d'adaptation des profils, dans la perspective à plus long terme d'enrichir l'interaction entre utilisateurs et système de filtrage.*

Nous adoptons un processus de formation des communautés d'utilisateurs qui exploite un algorithme de positionnement en 2 dimensions et un algorithme classique de classification non supervisée afin d'obtenir de véritables « cartes » des communautés. Ces cartes s'appuient sur deux critères différents de formation des communautés.

ABSTRACT. *Today, to deal with information overload, more and more efficient tools are needed for information retrieval. In order to get relevant information, users can rely on recommender systems that are based on user profiles. Nevertheless, a change of user interest is not always well accounted for, because of the passive role of users in the majority of existing systems. This paper presents the possibility of using "community maps" for the task of interactive profile adaptation in recommender systems.*

In order to produce these "community maps", we adopt a process based on a 2D positioning and a classical clustering algorithm. These maps account for two different criteria for user proximity.

MOTS-CLÉS : *Filtrage d'information, système de recommandation, interaction, communauté, cartes de communautés.*

KEYWORDS : *Information filtering, reccommender system, interaction, community, community maps.*

1. Introduction

1.1. *Systèmes de filtrage et interaction*

L'objectif principal des systèmes de filtrage adaptatif, ou système de recommandation, est d'envoyer des informations pertinentes (documents) aux utilisateurs tout en s'adaptant en permanence à leur besoin d'information. Pour cela, les moteurs de ces systèmes gèrent entre autres des profils d'utilisateurs permettant une meilleure sélection des documents. Les systèmes sont conçus pour adapter les profils au cours du temps en exploitant au mieux le *retour de pertinence* que les utilisateurs fournissent sur les documents envoyés (voir Figure 1).

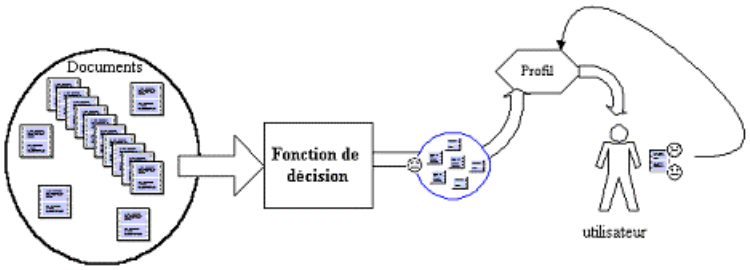


Figure 1. *Schéma de filtrage d'information.*

Il existe aujourd'hui de nombreux systèmes de recommandation appliqués à plusieurs domaines de la vie quotidienne comme le commerce électronique, les loisirs, la recherche documentaire scientifique, la gestion des connaissances, etc. On peut citer à titre d'exemple quelques sites Web populaires comme CiteSeer, Amazon, eBay, MovieLens, etc. Au-delà du domaine d'application, Montaner et al. présentent une taxonomie des systèmes de recommandation selon huit dimensions comprenant représentation et instanciation de profils, technique d'apprentissage, rétroactions et technique pour l'adaptation de profils, techniques de filtrage, de recoupement de profils, et de recoupement profil-document pour la production de recommandations (Montaner et al., 2003).

Le *retour de pertinence* que les utilisateurs fournissent aux systèmes prend généralement place dans un cadre contraignant : les utilisateurs perçoivent le système comme une boîte noire, et ils ne peuvent exprimer l'évolution de leur besoin d'information que sous la forme d'une succession d'évaluations des documents reçus. Ainsi, un changement dans le besoin d'information sera

potentiellement mal traduit par un utilisateur qui ne reçoit pas du système les documents lui permettant d'exprimer ce changement. L'utilisateur se trouvant dans l'impossibilité d'adapter son profil à son besoin, finira par abandonner le système, et cela quelle que soit la qualité du moteur de filtrage (Gallardo-Lopez et al., 2003).

1.2. *Positionnement des travaux présentés*

Dans les systèmes de filtrage collaboratif, le système envoie aux utilisateurs des documents ayant été jugés pertinents par d'autres utilisateurs ayant un point de vue proche (Herlocker et al., 2000). Pour ce faire, les systèmes regroupent virtuellement les utilisateurs en communautés, et facilitent les échanges de documents au sein d'une même communauté. Cependant, dans la plupart des systèmes de filtrage, les utilisateurs participent de façon passive à ces activités et le système ne se présente pas de façon transparente. Devant cette boîte noire, ils ne comprennent pas ce qui se cache derrière la formation de communautés réalisée par le système et comment eux-mêmes se situent au sein des communautés.

Nous pensons que les communautés sont un vecteur pour une interaction plus riche. En rendant les communautés visibles, perceptibles par les utilisateurs, nous envisageons à plus long terme de développer autour d'elles une interaction permettant à chaque utilisateur de comprendre, de modifier ou d'améliorer son profil via son positionnement au sein d'une communauté (Denos et al., 2004).

Voici un exemple de scénario : le système détecte une accumulation de retours de pertinence négatifs, ce qui traduit une situation susceptible de décourager un utilisateur U_0 ; le système entame un dialogue avec l'utilisateur en lui montrant sa position parmi les communautés, et en lui proposant de se rattacher à une communauté qui lui semble plus proche que celle dans laquelle il se situe actuellement. Le système peut par exemple permettre à l'utilisateur d'adopter le profil d'un utilisateur qui lui est proche.

Nous souhaitons étudier cette possibilité de rendre visibles les communautés aux utilisateurs et de leur permettre de se positionner dans ces communautés. Cela pose un certain nombre de problèmes : 1) comment calculer les communautés et assurer une certaine qualité dans ce calcul ; 2) sur quels critères définir la proximité entre utilisateurs sur laquelle fonder les communautés ; et 3) comment visualiser les communautés.

L'objectif principal de cet article est de répondre à ces trois questions : (1) calculer les communautés et les visualiser ; (2) l'appliquer à différents types de communautés : les communautés réunissant les personnes partageant les mêmes centres d'intérêt thématiques d'une part, les communautés partageant les mêmes avis sur les documents d'autre part, et (3) proposer d'utiliser l'algorithme des fourmis artificielles conjointement à l'algorithme des K-moyennes pour les visualiser en 2 dimensions.

Nous avons vérifié la faisabilité de notre approche sur de grandes quantités de données en des temps raisonnables.

1.3. *Plan de l'article*

Nous présentons dans la section 2 la notion de communauté dans les systèmes de filtrage, et nous décrivons ensuite dans la section 3 les méthodes sous-jacentes à leur génération : critères de formation, calculs de proximité entre utilisateurs, méthode et qualité de classification.

Ensuite, dans la section 4, nous présentons l'algorithme des fourmis artificielles, nous rappelons l'algorithme des K-moyennes et nous montrons l'application de ces algorithmes à la formation des communautés. La section 5 décrit l'application de cette proposition à un jeu de données réelles, « MovieLens data set »[1], ce qui permet d'en montrer la faisabilité et les performances en temps de calcul. Sur ces données, nous pouvons construire deux types de cartes qui sont ensuite analysées afin d'étudier les similarités et contrastes entre les cartes selon le critère de formation des communautés sur lequel elles s'appuient.

Nous donnons enfin une conclusion et les perspectives que ces résultats ouvrent pour des travaux futurs.

2. Communautés dans les systèmes de filtrage

2.1. *Notion de communauté dans les systèmes de filtrage*

Les communautés sont composées des utilisateurs qui sont proches les uns des autres relativement à un critère particulier : 1) critère basé sur le contenu, 2) critère basé sur les évaluations globales indépendamment du contenu. Selon le *critère de formation des communautés*, la position des utilisateurs et leur regroupement en communautés sont susceptibles de varier.

Nous pensons que les diverses formations de communautés, et les éventuels contrastes qu'elles sont susceptibles de faire surgir, peuvent servir de point de départ à l'adaptation du profil de l'utilisateur.

---

[1] http://www.cs.umn.edu/Research/GroupLens/

2.2. *Les communautés dans l'état de l'art*

Dans leur article (Perugini et al., 2003), les auteurs présentent un état de l'art sur des études liées à la communauté dans des systèmes de recommandation. A la différence du point de vue fonctionnel présenté dans (Montaner et al., 2003), Perugini et al. donnent un point de vue social sur les systèmes de recommandation, c'est-à-dire sous l'angle de l'effort pour établir des relations entre utilisateurs. Ces relations sont disponibles dans des profils-utilisateur (information explicite) ou bien au contraire, doivent être découvertes à partir de données véhiculant de façon implicite un réseau social, comme par exemple les affiliations de personnes que l'on peut trouver sur le Web (qui est affilié à quelle institution).

Par exemple, dans l'approche de fouille et d'exploration de structure (Mining and Exploiting Structure) (Mirza et al., 2003), le système transforme un réseau bipartite R (voir Figure 2), qui représente une matrice d'évaluations, ayant 2 classes de nœuds {personne p_i} et {document d_j}, en un réseau social unipartite G_s généralement en 3 étapes : fouiller le réseau ; identifier, modéliser/extraire le réseau social G_s, et rattacher les deux réseaux en G_r pour l'exploration et l'exploitation dans la production de recommandations. Dans leur article, les auteurs proposent la technique « hammock jump » qui relie deux personnes ayant w évaluations communes pour induire le réseau social.

Cette approche permet d'une part de reconnaître et d'explorer des structures dans l'ensemble des évaluations, et d'autre part de calibrer et d'évaluer la performance de système en termes de la connexion des utilisateurs avec des documents, en analysant des caractéristiques structurales des graphes Gs et Gr, par exemple, le nombre de personnes liées par la technique appliquée dans le système, le rapport entre le paramètre w et la taille de l'ensemble d'apprentissage qu'un utilisateur doit fournir au système pour recevoir des recommandations, etc.

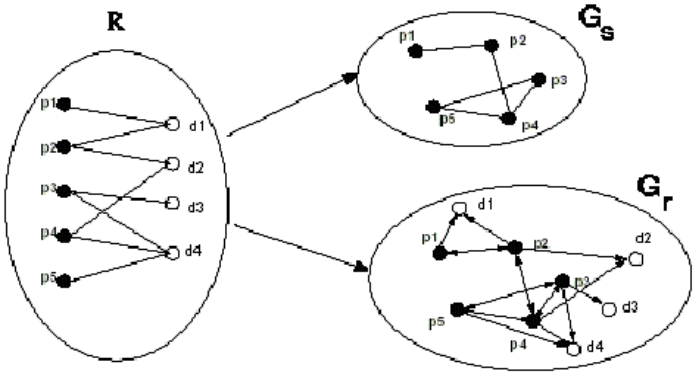


Figure 2. *Fouille et exploration de structure.*

Outre cette construction explicite d'un réseau social, les systèmes de filtrage collaboratif ne rendent que rarement explicites les communautés sur lesquelles s'appuient les recommandations : elles restent implicites dans la mesure de proximité entre utilisateurs.

3. Formation des communautés

Dans cette partie, nous discutons les divers aspects de la formation de communautés dans des systèmes de filtrage, c'est-à-dire les critères de formation, le calcul de proximité entre utilisateurs, les méthodes de classification et leur qualité.

3.1. *Critère de formation*

Les critères de formation explicite des communautés dans des systèmes de filtrage reposent sur :

– un critère de contenu, c'est-à-dire basé sur le contenu des documents que l'utilisateur a évalués. Nous l'appelons critère « contenu ».

– un critère qualitatif basé sur l'évaluation globale que l'utilisateur a donnée sur les documents. Ce dernier critère implique non seulement le contenu thématique mais aussi tous les autres critères de pertinence (fraîcheur, réputation de l'auteur, concision, etc.). C'est pourquoi nous l'appellerons critère « qualité ».

3.2. *Proximité entre utilisateurs*

Pour chaque critère de formation de communautés, il faut définir une *mesure de la proximité entre utilisateurs*. Les mesures les plus utilisées pour calculer de façon générale ou pour la formation de communautés sont les distances de Minkowski, Euclidienne, Manhattan, Chebychev, Mahalanobis, χ^2, Kullback-Leibler, la corrélation de Pearson, etc. (Jain et al., 1999).

On peut trouver de nombreuses études sur l'utilisation des distances dans différents domaines d'application comme la segmentation de documents audio, la reconnaissance des formes, la recherche d'images, etc. Cependant, il en existe peu concernant la formation de communautés dans les systèmes de filtrage. Dans ce contexte, on constate que la corrélation de Pearson est la plus utilisée (Breese et al., 1998 ; Herlocker et al., 1999).

Dans nos travaux, nous nous appuyons sur les mesures existantes en orientant notre choix par analogie avec les usages habituels.

3.3. *Classification*

Il faut également définir la *méthode de classification* afin de déterminer concrètement l'ensemble des communautés à partir de l'ensemble des utilisateurs. Nous présentons ici les méthodes générales susceptibles de s'appliquer au cas de la formation de communautés. On ne s'intéresse qu'aux méthodes de classification non supervisée (clustering) en raison du manque de connaissances a priori sur des modèles de communautés. Dans cet article, nous utilisons donc le terme « classification » pour désigner la classification non supervisée.

Parmi de nombreux algorithmes de classification existants, on peut citer trois algorithmes parmi les plus populaires : la classification ascendante hiérarchique (CAH), l'algorithme des K-moyennes, et l'algorithme des C-moyennes floues qui sont respectivement les représentants des trois approches de classification hiérarchique, de partitionnement, et de classification floue. Les lecteurs intéressés peuvent consulter d'autres algorithmes dans deux synthèses récentes (Berkhin, 2002) et (Jain et al., 1999). Outre ces techniques classiques, l'approche des fourmis artificielles (Ant Colony Optimization – ACO) permet de répartir une population sur un plan tout en reflétant la proximité entre les individus (Azzeg et al., 2004). C'est cet algorithme que nous utilisons dans notre approche et que nous justifions et expliquons dans la section 4.

3.4. *Qualité de classification*

Enfin, pour évaluer la qualité de la formation des communautés, on peut utiliser les critères classiques d'inerties intra-classe ([1]) et inter-classes ([2]) ou d'autres critères comme l'entropie spatiale ([3]) dans le cas d'un nombre important d'utilisateurs.

$$I_{intra}(C_j) = \frac{2}{|Cj|\cdot(|Cj|-1)} \sum_{x,y \in Cj} d^2(x,y) \qquad [1]$$

$$I_{inter}(C) = \sum_{j=1}^{k} \left[\frac{2}{|Cj|\cdot(|Cj|-1)} \sum_{x,y \in Cj} d^2(x,y) \right] \qquad [2]$$

$$E_S(C) = -\sum_{j=1}^{k} p(Cj).\log(p(Cj)) \qquad [3]$$

où $\quad P(C_j) = \dfrac{|Cj|}{nombre\ d'objets}$

3.5. *Etat des systèmes de filtrage*

Actuellement, on peut trouver des systèmes de filtrage hybrides qui combinent de diverses façons les deux critères « contenu » et « qualité » comme présentés dans (Burke, 2002). On retrouve donc en filigrane les critères de formation de communautés qui seront pris en compte dans cet article. Dans ces systèmes, on utilise souvent la corrélation de Pearson pour calculer la proximité entre utilisateurs.

4. Notre proposition : visualisation des communautés par l'algorithme des fourmis artificielles en combinaison avec une classification K-moyennes

Dans cette section, nous présentons les deux algorithmes des fourmis artificielles et des K-moyennes et notre proposition de les combiner pour la classification et la visualisation des communautés dans un système de filtrage.

4.1. *Algorithme des fourmis artificielles*

Dans un système de filtrage, les communautés doivent être aisément représentables et perceptibles par les utilisateurs. Nous proposons de les visualiser en 2 dimensions, visualisation que nous appelons « cartes de communautés ». Pour cela, nous nous intéressons à l'algorithme des fourmis artificielles qui a été proposé pour la première fois par Denebourg (Denebourg et al., 1990) pour le problème de tri d'objets.

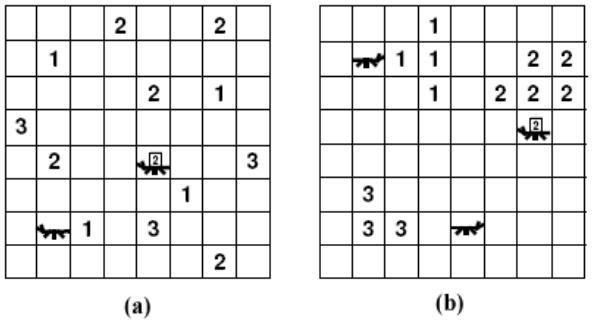


Figure 3. *Exemple de l'algorithme des fourmis artificielles (Azzag et al., 2004).*

L'idée principale de cet algorithme est que les objets correspondant à des points dans un espace à d dimensions *(d > 2)* sont plongés dans un espace à *2 dimensions*, c'est à dire une grille dont chaque cellule peut contenir un objet (voir Figure 3a) :

- Un agent (fourmi) n'a qu'une perception locale des objets et n'est pas capable de communiquer avec les autres ;
- Lorsqu'un agent libre rencontre un objet, il le ramasse avec une probabilité de $k_1/(k_1+f)$, où la fonction de densité f représente la proportion d'objets dans son voisinage et k_1 est une constante (voir Figure 3b);
- Une fois qu'un objet a été ramassé, l'agent chargé se déplace au hasard dans la grille et dépose l'objet avec une probabilité de f / (k2+f), où k2 est une constante. Il ne le dépose que si cette probabilité dépasse un certain seuil (voir Figure 3b).

En 1994, Lumer et al. ont apporté des modifications (Lumer et al., 1994), notamment en remplaçant la fonction de densité f par une moyenne des similarités entre l'objet en question et les objets situés dans son voisinage ([4]).

L'algorithme d'origine de Denebourg est donc devenu un véritable algorithme de classification (voir Figure 4), et on trouve de nombreuses études et applications sur cette classification biomimétique (Azzag et al., 2004 ; Dorigo et al., 2000).

$$f(x_i) = \max\left(\frac{1}{s^2}\sum_{x\in V(x_i)}\left[1-\frac{d(x,x_i)}{\alpha}\right], 0\right) \qquad [4]$$

où s^2 : taille de voisinage V et

α : constante

Les avantages de cet algorithme sont 1) la possibilité de l'appliquer à plusieurs types de données, et 2) la possibilité de visualiser le résultat aisément.

En ce qui concerne l'évaluation de performance algorithmique, Handl et al. ont présenté leur comparaison de l'algorithme des fourmis artificielles modifié (Ant-based Clustering) avec les 3 autres algorithmes : K-moyennes, classification ascendante hiérarchique et 1D-SOM (One-dimensional Self-Organising Maps) sur 6 jeux de données dont 3 jeux artificiels et 3 jeux réels provenant de UCI Knowledge Discovery in Databases Archive[2], University of California (Handl et al., 2003). En utilisant quatre critères comme F_Measure, Rand Index, Inner Cluster Variance et Dunn Index, les auteurs ont observé que l'algorithme de fourmis artificielles donne d'excellents résultats.

[2] http://kdd.ics.uci.edu

```
LF_AntBasedClustering (Objets = {x1,…, xn}, Agents = {a1,…, aA}, grille G)
{   (S0) INITIALISATION : Placer aléatoirement les objets et les agents sur G.
    (S1) BOUCLE PRINCIPALE
        for (t = 1; t <= tmax ; t++)
            for (ag = 1; ag <= nombre d'agents; ag++)
            {
                if (l'agent [ag] ne transporte pas d'objet et trouve un objet [xi])
                    if  (la probabilité de ramasser probPick(ag, xi) >= seuil)
                        pickItem(ag, xi)
                else if (l'agent [ag] transporte un objet [xi] et la cellule est vide)
                    if (la probabilité de déposer probDrop(ag, xi) >= seuil)
                        dropItem(ag, xi);
                L'agent [ag] se déplace vers une cellule non occupée
            }
    (S2) return (l'emplacement des objets {x1,…, xn} sur la grille G)
}
```

Figure 4. *Algorithme de fourmis artificielles de Lumer et al.*

4.2. *K-moyennes*

L'idée principale de l'algorithme des K-moyennes est de classifier des objets en k classes en minimisant la variance intra-classe et en maximisant l'écartement inter-classes (voir Figure 6). Cet algorithme choisit d'abord au hasard des centres de gravité et construire les classes initiales autour de ces centres (Etape 0-a et Etape 0-b dans la Figure 5). Dans chaque itération, on recalcule les centres et forme les nouvelles classes (Etape 1-a et Etape 1-b dans la Figure 5), et finalement, le processus se termine quand on n'obtient plus de changement de partition.

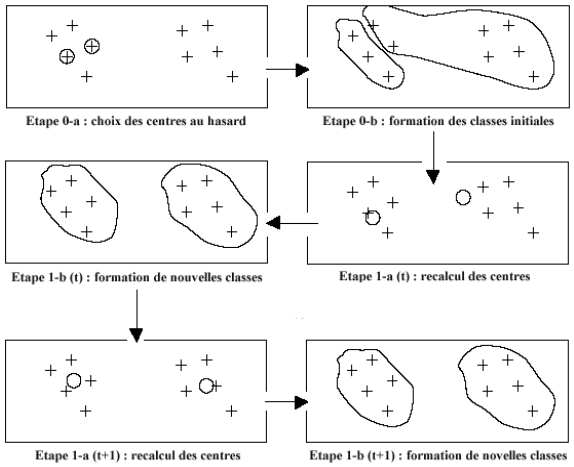


Figure 5. *Exemple de l'algorithme des K-moyennes.*

K_means (Objets = $\{x_1,\ldots, x_n\}$, k)

{ (S0) INITIALISATION :

a) Choisir au hasard k centres de gravité : $G^{(0)} = \{g_1,\ldots,g_k\}$

b) Construire la partition de k classes : $C^{(0)} = \{C_1,\ldots, C_k\}$

où $C_j = \{ x \in \text{Objects} / \forall i \neq j, d(x,g_j) < d(x,g_i) \}, \forall j \in [1,k] \}$

(S1) BOUCLE PRINCIPALE

a) Recalculer les centres : $G^{(t)} = \{g_1,\ldots,g_k\}$

où $g_j = \frac{\sum_{x \in Cj} x}{|Cj|}, \forall j \in [1,k]$ [5]

b) et les classes : $C^{(t)} = \{C_1, \ldots, C_k\}$

Test d'arrêt : ($C^{(t+1)} \cong C^{(t)}$) ou (nombre d'itération > seuil)

(S2) **return** (la partition $C^{(t)} = \{C_1, \ldots, C_k\}$)

}

Figure 6. *Algorithme des K-moyennes.*

4.3. *Application aux communautés dans un système de filtrage*

Nous proposons de combiner les deux algorithmes ci-dessus pour la classification et la visualisation des communautés dans un système de filtrage (voir Figure 7). Nous appliquons d'abord l'algorithme des fourmis artificielles pour positionner les utilisateurs dans un espace en 2 dimensions (1), et la carte de communautés est finalement obtenue par l'application de l'algorithme des K-moyennes sur le positionnement résultant du premier processus (2). Nous avons choisi ces deux algorithmes en fonction de la capacité de visualiser aisément les résultats du premier algorithme et en raison de la simplicité et l'efficacité du deuxième algorithme dans la plupart des cas (Jain et al., 1999).

Dans le schéma de fonctionnement ci-dessus, nous avons besoin d'une mesure de proximité appliquée entre les profils utilisateur pour alimenter le processus de positionnement des utilisateurs dans l'algorithme des fourmis artificielles (3). Plus précisément, cette mesure est utilisée pour le calcul de la formule [4]. Sachant que nous nous intéressons à des systèmes hybrides qui exploitent à la fois le contenu et la qualité des documents, nous utilisons de fait deux mesures de proximité : 1) la corrélation de Pearson pour le calcul sur le critère « qualité », par analogie avec les systèmes de filtrage collaboratif, 2) pour le critère « contenu », nous choisissons la distance Euclidienne en raison de sa haute performance dans les espaces euclidiens en 2 dimensions. Ceci a été montré par des travaux expérimentaux (Jain et al., 1999).

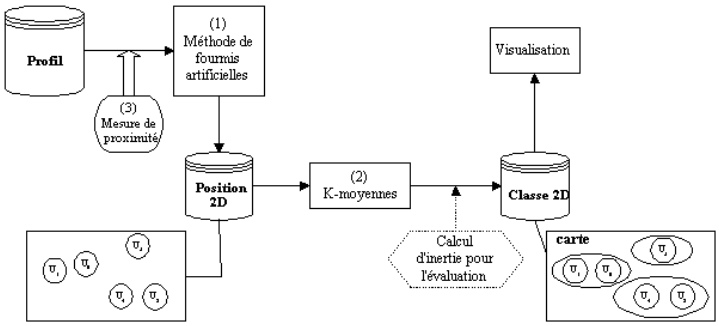


Figure 7. *Schéma de fonctionnement de l'approche proposée.*

5. Application au jeu de données « MovieLens data set »

Après avoir présenté notre approche dans la section 4, nous donnons sa justification en termes de performance (temps d'exécution) et décrivons la façon d'analyser la similarité entre les cartes de communautés « contenu » et « qualité ».

Nous présentons d'abord quelques chiffres sur le jeu de données réelles MovieLens et sur les calculs pour la construction des profils. Ensuite, nous montrons la performance des algorithmes appliqués et finalement, nous présentons les mesures et les résultats d'analyse des cartes de communautés.

5.1. *Données et critères de formation des communautés*

Afin de justifier notre approche, nous avons utilisé la base de données réelles MovieLens[3] issue du groupe de recherche GroupLens à l'université du Minnesota, qui contient 100 000 évaluations sur 1 682 films/vidéo réalisées par 943 personnes, dont le score est donné sur une échelle allant de 1 à 5 étoiles.

Conformément aux objectifs de nos travaux, les deux critères de formation des communautés (contenu et qualité) peuvent être extraits de ces données. Notre but est de définir, pour chaque critère, les profils d'utilisateurs qui serviront au calcul de proximité. Nous les définissons comme suit.

Pour le critère « contenu », la base propose 19 genres de film, et chaque film peut être associé à plusieurs genres à la fois. Par exemple, certains films sont associés à 6 genres. Pour chaque utilisateur, nous construisons la partie « contenu » de son profil comme un vecteur de 19 valeurs (voir Figure 8) qui reflètent le niveau d'intérêt qu'il a manifesté pour les genres de film, en se basant sur le nombre, la moyenne et la variance de ses évaluations concernant chacun des genres.

	action	adventure	animation	children	comedy	crime	documentary	drama	…	war	western
U1	5,86	4,99	4,83	3,83	6,30	5,19	5,72	6,90	…	5,27	4,73
U2	7,05	6,51	5,46	5,49	7,42	6,99	0,00	9,16	…	6,31	0,00
…											

Figure 8. *Extrait des profils « contenu » (poids en %).*

Pour l'aspect « qualité », le profil de chaque utilisateur comprend toutes ses évaluations.

Quelques statistiques simples permettent d'avoir un premier éclairage sur ces données : 1) le nombre d'évaluations faites par chaque personne varie de 20 à 737 ; et 2) on constate une proportion faible de scores défavorables (6,11% et 11,37%

[3] http://www.grouplens.org

pour 1 et 2 étoiles respectivement) qui montre la tendance des utilisateurs à n'évaluer que les films qu'ils aiment.

5.2. *Performance*

Nous avons appliqué la chaîne des processus présentée dans la Figure 7 aux profils « contenu » ainsi qu'aux profils « qualité ». Nos résultats présentés dans le reste de cet article sont obtenus à partir de 50 essais indépendants réalisés sur un PC de 2,40Ghz de processeur et 1Go de mémoire.

5.2.1. Performance pour l'algorithme des fourmis artificielles

Pour calibrer les paramètres de l'algorithme des fourmis artificielles, nous avons exploité les expériences de Handl (Handl et al., 2003) et de Dorigo (Dorigo et al., 2000) avec quelques modifications afin d'obtenir de bons résultats (voir Figure 3 et Tableau 1).

Paramètre	Valeur
k_1	0,10
k_2	0,15
alpha (carte contenu)	0,50
alpha (carte qualité)	1,40

Paramètre (N=943 personnes)	Valeur
Taille de grille ($\sim\sqrt{N.10}$)	100
Nombre d'agents	10
Taille de voisinage	3 x 3
Seuil de ramasser et de déposer	0,05

Tableau 1. *Valeurs de paramètre pour l'algorithme des fourmis artificielles.*

En se basant sur l'inertie ([2]), nous avons constaté que l'algorithme des fourmis artificielles donne de bonnes classifications à partir de 1,5 millions d'itérations (voir Tableau 2), et nous avons choisi deux millions d'itérations (~ 2000.N), comme Handl et al. L'ont proposé, compte tenu du d'exécution raisonnable (moins de 3mn).

	5×10^5	10×10^5	15×10^5	20×10^5	25×10^5	30×10^5
Inertie	0,218415	0,202920	0,168476	0,166612	0,166775	0,162675
Temps[4] (s)	48,53	88,94	129,53	172,72	217,69	256,34

Tableau 2. *Inertie des classes 2D finales selon le nombre d'itérations choisi pour l'algorithme des fourmis artificielles (k = 10).*

[4] sans compter le temps de calcul des distances entre utilisateurs.

5.2.2. Performance pour l'algorithme des K-moyennes

En appliquant l'algorithme des K-moyennes sur le résultat de premier processus, la vitesse de convergence est assez rapide du fait que les utilisateurs sont déjà bien regroupés par l'algorithme des fourmis artificielles. On peut obtenir des classifications stables après une moyenne de vingtaine d'itérations et moins de 10 secondes (voir Tableau 3).

	K = 10	K = 15	K = 19
Itération	18,16	19,84	20,92
Temps	7,90	8,60	9,56

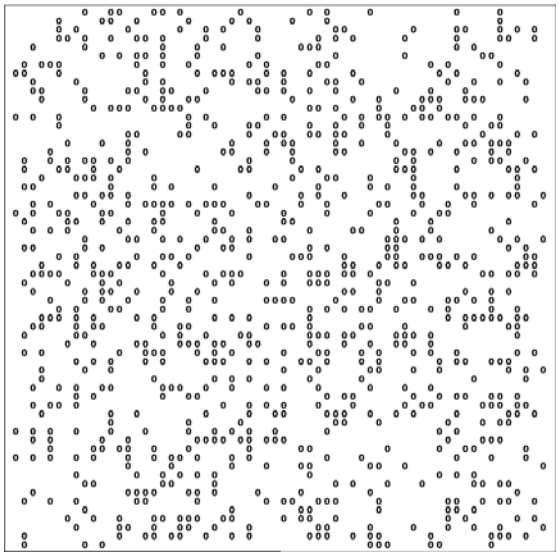
Tableau 3. *Performance de l'algorithme des K-moyennes.*

5.3. *Comparaison de cartes de mesures*

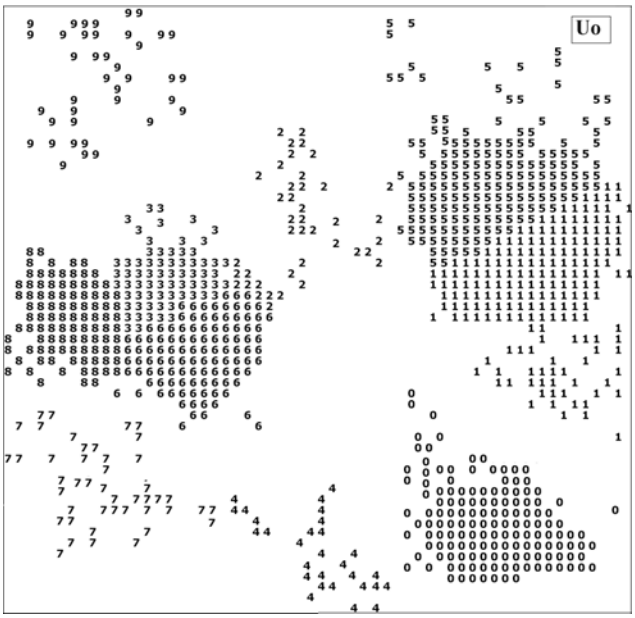
5.3.1. Mesures d'analyse

L'objectif de l'analyse est de comparer les cartes « contenu » avec les cartes « qualité ». On peut les comparer visuellement « à l'œil nu » (voir Figure 9) pour une première impression.

(Figure 9a)
Carte initiale



(Figure 9b)
Carte
« contenu »



(Figure 9c)
Carte
« qualité »

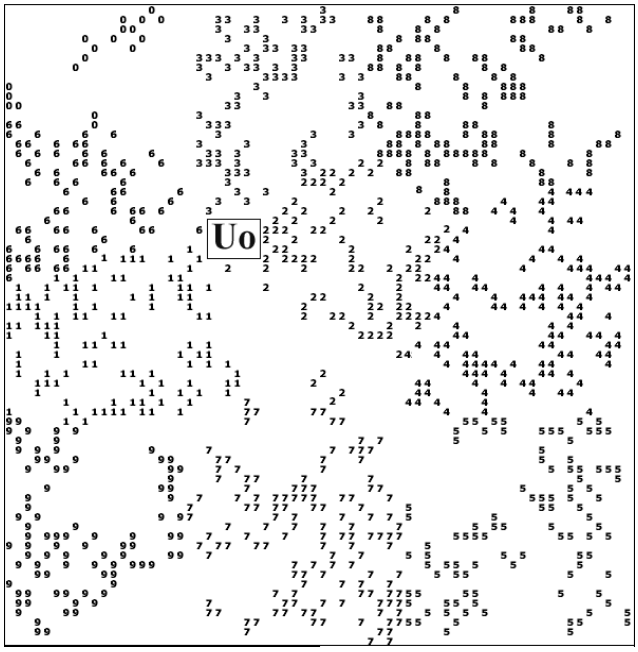


Figure 9. *Visualisation des cartes de communautés construites à partir des données réelles de MovieLens (k=10).*

Pour obtenir des conclusions statistiques, nous utilisons les indicateurs Rand Index et F_Measure, comme dans les expériences de Handl et al., qui permettent de mesurer le niveau de contraste de ces cartes.

a) Rand Index : $$R = \frac{a+d}{a+b+c+d} \quad [6]$$

où $a = \{x_i, x_j / classe_{cont}(x_i) = classe_{cont}(x_j)$ and $classe_{qual}(x_i) = classe_{qual}(x_j)\}$

$b = \{x_i, x_j / classe_{cont}(x_i) = classe_{cont}(x_j)$ and $classe_{qual}(x_i) \neq classe_{qual}(x_j)\}$

$c = \{x_i, x_j / classe_{cont}(x_i) \neq classe_{cont}(x_j)$ and $classe_{qual}(x_i) = classe_{qual}(x_j)\}$

$d = \{x_i, x_j / classe_{cont}(x_i) \neq classe_{cont}(x_j)$ and $classe_{qual}(x_i) \neq classe_{qual}(x_j)\}$

b) F_Measure : $$F = \frac{2 . p_{ij} . r_{ij}}{(p_{ij} + r_{ij})} \quad [7]$$

où n_i : nombre d'éléments d'une classe C_i dans la carte « contenu »

n_j : nombre d'éléments d'une classe C_j dans la carte « qualité »

n_{ij} : nombre d'éléments de la classe C_i apparaissant dans la classe C_j

$p_{ij} = n_{ij} / n_j$ et $r_{ij} = n_{ij} / n_i$

5.3.2. Résultats

On peut aisément constater un écartement significatif entre les cartes de communautés prises en exemple dans la Figure 9. Les personnes dans la carte « contenu » sont bien regroupées par rapport à la carte « qualité ». L'écartement entre ces cartes est aussi reflété par les indicateurs Rand Index et F_Measure dans le Tableau 4 et le Tableau 5. Les valeurs faibles dans ces deux tableaux montrent la différence nette et permanente entre les deux cartes « contenu » et « qualité ».

Indication de similarité	K = 10	K = 15	K = 19
Rand Index	0,213784	0,150734	0,119748
F_Measure	0,163473	0,133683	0,121261

Tableau 4. *Faible similarité des cartes « contenu » et « qualité ».*

Indication	11/97	12/97	01/98	02/98	03/98	04/98
Rand Index	0,064818	0,090343	0,120032	0,120032	0,194652	0,213784
F_Measure	0,187740	0,185326	0,180546	0,177703	0,174561	0,163473

Tableau 5. *Analyse mensuelle des indications Rand Index et F_Measure : permanence de la faible similarité entre les cartes « contenu » et « qualité ».*

5.4. *Synthèse de l'expérimentation*

En résumé, on constate sur le jeu de données de MovieLens des contrastes entre la carte traduisant la dimension de « contenu » et la carte traduisant la dimension de « qualité », contrastes que l'on espère pouvoir exploiter comme point de départ à un diagnostic de la situation de l'utilisateur, ou au moins à l'adaptation du profil de l'utilisateur.

Ainsi, reprenons le scénario donné en introduction où l'utilisateur U_0 est en situation difficile, reflétée par son envoi de feedbacks négatifs. On peut voir dans la carte « contenu » que cet utilisateur est très isolé (voir Figure 10a) tandis qu'il se positionne à l'intersection de plusieurs classes dans la carte « qualité » (voir Figure 10b). Voici les cartes que l'on pourrait proposer à l'utilisateur U_o pour lui demander de se rattacher à une des classes représentées par des utilisateurs typiques symbolisés par des cercles sur la carte « qualité ».

6. Conclusion et perspectives

6.1. *Bilan des travaux présentés*

Dans cet article, nous avons proposé une approche (les « cartes de communautés ») constituant une première étape dans la perspective d'améliorer les systèmes de filtrage en enrichissant l'interaction entre utilisateurs et système de filtrage en particulier dans la tâche d'adaptation des profils.

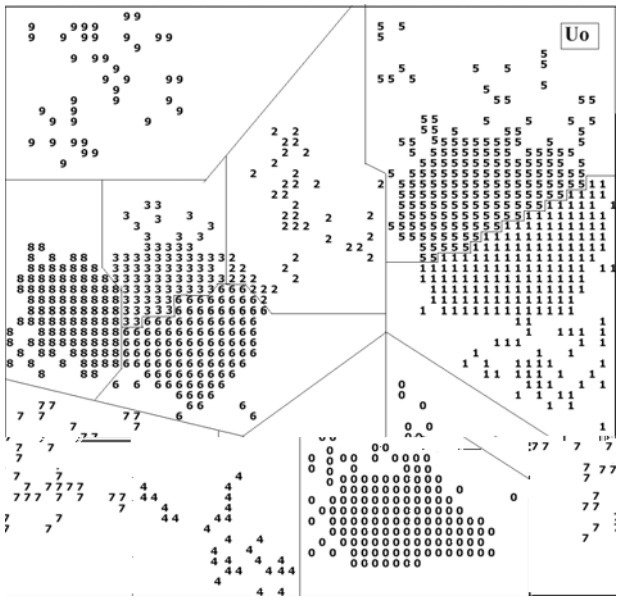
Nous avons montré par l'expérience que la production des cartes de communautés est faisable via une combinaison des algorithmes de fourmis artificielles, qui mesure la proximité entre utilisateurs et les positionne en 2D, et cette expérience rend compte d'un temps de calcul raisonnable.

6.2. *Perspectives*

Dans les travaux futurs, nous envisageons d'ajouter d'autres critères de formation des communautés. En effet, dans l'approche actuelle les communautés sont seulement construites selon les vues que le système peut établir sur les utilisateurs (profil « contenu » et profil « qualité »). Nous pouvons envisager d'exploiter également, par exemple, les centres d'intérêt des utilisateurs tels qu'ils les formulent eux-mêmes explicitement. Ce type d'interaction trouvera naturellement sa place dans une plate-forme de filtrage collaboratif orientée vers la

communauté comme COCoFil (Denos et al., 2004), où les utilisateurs décrivent leur propre perception de leur communauté via leur carnet d'adresse.

(Figure 10a)
Carte « contenu »



(Figure 10b)
Carte « qualité »

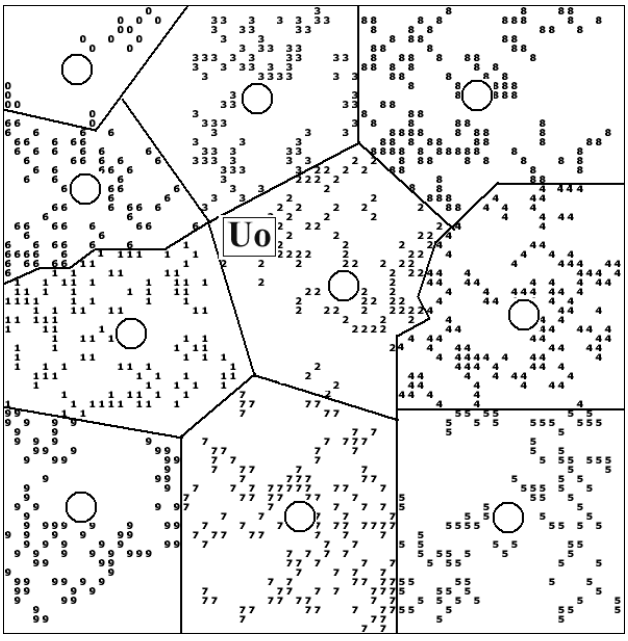


Figure 10. *Proposition de positionnement dans des communautés.*

Nous souhaitons également concevoir des processus d'adaptation du profil selon les situations rencontrées par les utilisateurs. Ceci nécessite une base de connaissances sur les situations d'interaction consistant par exemple en la cause, le but, les critères et les acteurs de la formation des communautés (système/utilisateur).

Finalement, cet article nous amène aussi à étudier l'évolution des communautés, sur l'hypothèse que l'analyse temporelle des cartes de communautés permettra de construire une méthode d'apprentissage sur le modèle de similarités et de contrastes entre cartes de communautés. Cette analyse pourra servir à la tâche de détection du changement d'intérêt des utilisateurs.

7. Bibliographie

Azzag H., Picarougne F., Guinot C., Venturini G., Un survol des algorithmes biomimétiques pour la classification, *Classification et fouille de données*, RNTI-C-1, Cépaduès, 2004.

Berkhin P., « Survey of Clustering Data Mining Techniques », Technical Report, Accrue Software, 2002.

Breese J. S., Heckerman D., Kadie C., « Empirical Analysis of Predictive Algorithms for Collaborative Filtering », *The 14th Conference on Uncertainty In Artificial Intelligence (UAI'98)*, July 1998**,** Madison, Wisconsin, USA, p.43-52.

Burke R., « Hybrid Recommender Systems: Survey and Experiments », *Journal of Personalization Research, User Modeling and User-Adapted Interaction*, vol. 12 (4), 2002, Kluwer Academic Publishers, p.331-370.

Deneubourg J.-L., Goss S., Franks N., Sendova-Franks A., Detrain C., Chrétien L., « The dynamics of collective sorting: robot-like and ant-like robots », *The 1st International Conference on Simulation of Adaptive Behavior (SAB'90)*, 1990, Massachusetts, USA, p.356-365.

Denos N., Berrut C., Gallardo-Lopez L., Nguyen A-T., « COCoFil : Une plateforme de filtrage collaboratif orientée vers la communauté », *Actes de la 1ère Première COnférence en Recherche d'Information et Applications (CORIA'04),* Mars 2003, Toulouse, France, p.9-26.

Dorigo M., Bonabeau E., Theraulaz G., Ant algorithms and stigmergy, *Future Generation Computer Systems (FGCS)*, vol. 16, Elsevier, 2000, p.851-871.

Gallardo-Lopez L., Berrut C., Denos N., « Une approche pour le contrôle de la qualité des Systèmes de Filtrage Collaboratif », *Manifestation de Jeunes Chercheurs STIC (MAJESTIC'03)*, Octobre 2003, Polytechnique de Marseille, France.

Handl J., Knowles J., Dorigo M., « On the performance of ant-based clustering », *The 3rd International Conference on Hybrid Intelligence Systems*, December 2003, Australia.

Herlocker L. J., Konstan J. A., Riedl J., Explaining Collaborative Filtering Recommendations, *The 2000 ACM Conference on Computer Supported Cooperative Work (CSCW'00)*, December 2000, Pennsylvania, USA, p.241-250.

Herlocker L. J., Konstan A. J., Borchers A., Riedl J., An Algorithmic Framework for Performing Collaborative Filtering, *The 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, USA, August 1999, p.230-237.

Jain A. K., Murty M. N., Flynn P. J., « Data Clustering: A Review », *ACM Computing Surveys*, vol. 31 (3), 1999, p.264-323.

Lumer E., Faieta B. « Diversity and Adaptation in Populations of Clustering Ants », *From Animals to Animats 3: The 3rd International Conference on Simulation of Adaptive Behavior (SAB'94)*, 1994, p.501-508.

Mirza B. J., Keller B. J., Ramakrishnan N., « Studying Recommendation Algorithms by Graph Analysis », *Journal of Intelligent Information Systems*, vol. 20 (2), Mar 2003.

Montaner M., López B., De La Rosa J. L., « A Taxonomy of Recommender Agents on the Internet », *Artificial Intelligence Review*, vol. 19, 2003, Kluwer Academic Publishers, p.285-330.

Perugini S., Gonçalves M. A., Fox E. A., « A Connection-Centric Survey of Recommender Systems Research », *Journal of Intelligent Information Systems*, vol. 23 (1), 2003.